

VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky

BAKALÁŘSKÁ PRÁCE

2018

Ondřej Laboj

VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra kybernetiky a biomedicínského inženýrství

Autonomní řízení modelu auta
Autonomous Control of the Car Model

2017 / 2018

Ondřej Laboj

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra kybernetiky a biomedicínského inženýrství

Zadání bakalářské práce

Student: **Ondřej Laboj**
Studijní program: **B2649 Elektrotechnika**
Studijní obor: **2612R041 Řídicí a informační systémy**
Téma: **Autonomní řízení modelu auta**
Autonomous Control of the Car Model
Jazyk vypracování: **čeština**

Zásady pro vypracování:

Cílem bakalářské práce je návrh a implementace autonomního řízení modelu auta. Základem pro práci jsou modely a komponenty využité pro soutěž NXP Cup. Cílem práce je vytvořit funkční konkurenceschopný model s využitím ve výše zmíněné soutěži.

V souhrnu je práce charakterizována těmito body:

1. Řešení autonomního řízení modelu.
2. Návrh a realizace hardwarové části modelu.
3. Návrh a implementace řídicího softwaru.
4. Návrh, realizace a porovnání regulačních algoritmů pro optimalizaci provozu modelu.
5. Test modelu na reálné trati.
6. Zhodnocení práce a závěr.

Seznam doporučené odborné literatury:


- [1] MAZIDI, Muhammad A., NAIMI, Sarmad, NAIMI, Sepher a Shujen CHEN. *Freescale ARM Cortex-M Embedded Programming*. 2 edition. {s.l.}: MicroDigitalEd, 2016. ISBN 978-0997925982.
- [2] SCARPINO, Matthew. *Motors for Makers: A Guide to Steppers, Servos, and Other Electrical Machines*. 1 edition. Indianapolis: Que Publishin, 2015. ISBN 978-0134032832.
- [3] KIM, Sang-Hoon. *Electric Motor Control: DC, AC, and BLDC Motors*. Oxford (UK): Elsevier Science, 2017. ISBN 978-0128121382.
- [4] BALÁTĚ, Jaroslav. *Automatické řízení*. Praha: BEN - technická literatura, 2004. ISBN 978-80-7300-148-3.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

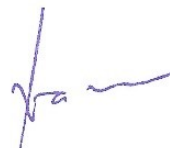
Vedoucí bakalářské práce: **Ing. Zdeněk Slanina, Ph.D.**

Datum zadání: 01.09.2017

Datum odevzdání: 30.04.2018



doc. Ing. Jiří Koziorek, Ph.D.
vedoucí katedry

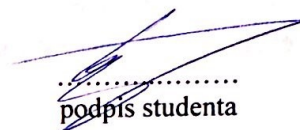


prof. Ing. Pavel Brandštetter, CSc.
děkan fakulty

Prohlášení studenta

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě dne: *30. dubna 2018*



.....
podpis studenta

Poděkování

Rád bych poděkoval Ing. Tomášovi Dočkalovi za vstřícnost při konzultacích a cenné rady k řešení problematiky práce. Dále děkuji vedoucímu bakalářské práce Ing. Zdeňkovi Slaninovi, Ph.D. za metodickou, pedagogickou, odbornou pomoc a konzultaci při vytváření této bakalářské práce. V neposlední řadě bych chtěl poděkovat přítelkyni, rodině a přátelům za podporu a motivaci.

Abstrakt

Cílem bakalářské práce je vytvořit autonomně řízený model auta, který se zúčastní soutěže NXP Cup. Nejdůležitějším faktorem a problematikou je rychlost auta, protože musí co nejrychleji projet závodní dráhu. U hardwarových úprav vozidla se moc nedá inovovat, protože NXP Cup má striktní podmínky na podvozek vozidla a další jeho části. Rozhodujícím faktorem rychlosti je tedy naprogramování modelu auta. Důležité je vyřešit přesnou komunikaci mezi kamerou a mikrokontrolérem, který se stará o řízení celého modelu auta. O pohon se starají dva motorky na bázi stejnosměrného proudu. K řešení práce je potřeba cvičná dráha, na které se bude testovat rychlost, přesnost a hbitost autíčka. NXP Cup je celosvětově uznávaná soutěž, ve které chci díky této práci dosáhnout skvělých výsledků a reprezentovat tak VŠB.

Klíčová slova

NXP Cup, kamera, mikrokontrolér, motorky s pohonem stejnosměrného proudu, podvozek, Freescale, FRDM-KL25Z, servomotor

Abstract

Purpose of this bachelor thesis is to create an autonomously controlled model of a car that will participate in the NXP Cup competition. The most important factor and problem is speed of the car because it must pass the race track as fast as possible. The hardware modifications of the car are not very innovative because the NXP Cup has strict conditions for the vehicle's chassis and other parts. The decisive factor of speed is the way how car model will be programmed. It is important to solve the exact communication between the camera and the microcontroller that takes care of the whole car model. Two DC-engines are powered by the drive. To solve the work, a training track is required to test the speed, accuracy and agility of the car. My own personal goal is to successfully represent technical university of Ostrava and achieve excellent results in the NXP Cup which is the world-renowned competition.

Key words

NXP Cup, camera, microcontroller, DC-motors, chassis, Freescale, FRDM-KL25Z, servomotor

Obsah

| | |
|---|--------|
| Seznam ilustrací | - 9 - |
| Seznam tabulek | - 10 - |
| Úvod..... | - 13 - |
| 1 Rešerše autonomního řízení modelu | - 14 - |
| 1.1 Požadavky na model auta | - 14 - |
| 1.2 Pravidla soutěže..... | - 15 - |
| 1.3 Popis tratě..... | - 15 - |
| 1.4 Analýza problematiky | - 16 - |
| 1.5 Světové trendy v autonomním řízení modelů aut..... | - 16 - |
| 2 Návrh a realizace hardwarové části modelu | - 17 - |
| 2.1 Freescale FRDM-KL25Z | - 17 - |
| 2.2 TFC Shield | - 18 - |
| 2.3 Podvozek | - 19 - |
| 2.4 DC motory..... | - 20 - |
| 2.4.1 Princip DC motoru | - 20 - |
| 2.5 Servomotor | - 22 - |
| 2.5.1 Řídicí elektronika | - 22 - |
| 2.5.2 Servomotor Futaba S3010 | - 22 - |
| 2.6 Baterie | - 23 - |
| 2.7 Kamera | - 24 - |
| 3 Návrh a implementace řídicího softwaru | - 25 - |
| 3.1 Analogově-digitální převodník..... | - 26 - |
| 3.2 Detekce obrazu | - 26 - |
| 3.3 Ovládání řízení | - 28 - |
| 3.3.1 Zpracování signálu PWM..... | - 28 - |
| 3.4 Detekce polohy..... | - 29 - |
| 3.4.1 Popis řešení v programu | - 29 - |
| 3.5 Nastavení motorů | - 30 - |
| 3.6 Nastavení tlačítek | - 30 - |

| | | |
|-------|---|------|
| 3.7 | Nastavení LED | 30 - |
| 3.8 | Hlavní smyčka..... | 31 - |
| 4 | Návrh, realizace a porovnání regulačních algoritmů pro optimalizaci provozu modelu..... | 32 - |
| 4.1 | Identifikace motorů | 32 - |
| 4.1.1 | Popis programu v Arduinu | 32 - |
| 4.1.2 | Vlastnosti magnetického snímače a kroužku..... | 33 - |
| 4.1.3 | Výpočet naměřené maximální lineární rychlosti..... | 34 - |
| 4.1.4 | Popis programu v Matlabu | 35 - |
| 5 | Test modelu na reálné trati | 36 - |
| 5.1 | Soutěžní kolo NXP Cup | 36 - |
| 6 | Zhodnocení práce a závěr..... | 37 - |
| | Použitá literatura | 38 - |
| | Seznam příloh..... | 40 - |

Seznam použitých symbolů a zkratek

| Zkratka | Význam |
|--------------|---|
| ABS | Akrylonitril-butadien-styren (Amorfní termoplastický kopolymer) |
| ADC | Analog to Digital Convertor (Analogově číslicový převodník) |
| AO | Analog output (Analogový výstup) |
| CFG | Configuration register (Konfigurační registr) |
| CLK | Clock (Hodiny) |
| GPIO | General-purpose input/output (obecné účely pro vstup/výstup) |
| I2C | Inter-integrated circuit (Multi-master počítačová sériová sběrnice) |
| LED | Light-emitting diode |
| Li-ON | Lithium-Iontová baterie |
| MOD | Modulo |
| NiCD | Nikl-kadmová baterie |
| NiMH | Nikl-metal hydridová baterie |
| PCOR | Pin Control Register |
| PDDR | Port data direction register |
| PDIR | Port Data Input Register |
| PSOR | Port Set Output Register (Port |
| PWM | Pulse width modulation (Pulzně šířková modulace) |
| RGB | Red-green-blue (Červená-zelená-modrá) |
| SI | Start impuls |
| SOPT | System options register (Systémové možnosti registru) |
| SPI | Serial peripheral interface (Sériové periferní rozhraní) |
| SRAM | Static random access memory (Statická paměť) |
| TPM | Timer/PWM module (Časovač/PWM modul) |
| UART | Universal a/synchronous receiver (A/Synchronní sériové rozhraní) |
| USB | Universal serial bus (Univerzální sériová sběrnice) |

Seznam ilustrací

| | |
|---|------|
| Obrázek 1. Základní rozdíly mezi modely [2]..... | 14 - |
| Obrázek 2. Příklad závodní dráhy [3]..... | 16 - |
| Obrázek 3. Popis částí modelu auta..... | 17 - |
| Obrázek 4. Ukázka TFC-shieldu na Freescale desce [4]..... | 18 - |
| Obrázek 5. Podvozek Alamak..... | 19 - |
| Obrázek 6. Elektromagnet [6]..... | 20 - |
| Obrázek 7. Rotace elektromagnetu [6]..... | 21 - |
| Obrázek 8. Popis DC motoru [6]..... | 21 - |
| Obrázek 9. Servomotor Futaba S3010 [10]..... | 23 - |
| Obrázek 10. Umístění baterie na podvozku..... | 24 - |
| Obrázek 11. Použitá kamera na modelu auta..... | 24 - |
| Obrázek 12. Vývojový diagram programu..... | 25 - |
| Obrázek 13. Integrační čas..... | 26 - |
| Obrázek 14. Synchronizace signálu SI a CLK pro získání AO [13]..... | 27 - |
| Obrázek 15. PWM cyklus pro servomotor..... | 28 - |
| Obrázek 16. Určení středu pozice na dráze..... | 29 - |
| Obrázek 17. Zapojení pro identifikaci motorů..... | 32 - |
| Obrázek 18. Správné umístění prstence [16]..... | 33 - |
| Obrázek 19. Přechodová charakteristika identifikovaného přenosu a naměřených dat..... | 35 - |
| Obrázek 20. Kvalifikační tabulka v kategorii Alamak [17]..... | 36 - |

Seznam tabulek

| | |
|--|--------|
| Tabulka 1. Specifikace motoru [7]..... | - 20 - |
| Tabulka 2. Specifikace servomotoru Futaba [9]..... | - 22 - |
| Tabulka 3. Srovnání baterii [11] [12] | - 23 - |

Úvod

Od vzniku prvního automobilu v 18. století lidé nesmírně pokročili ve výrobě aut a jejich možnostech. Staly se tak nedílnou součástí našeho života a v dnešní době si málokdo dokáže představit život bez dopravních prostředků. Obrovský pokrok automobilový průmysl zaznamenal před pár lety, kdy se začalo inovovat v autonomnosti vozidel. Už se nejedná o sny a představy našich předchůdců, ale opravdu se začaly vyrábět soběstačně řízené automobily, které mění historii a vytváří budoucnost automobilového průmyslu. Za nehody na silnicích může právě člověk a málokdy auto samo o sobě. Lidskou nedokonalost v řízení automobilu by proto mohly vyřešit autonomní vozidla. Například automobilka Tesly tato auta vyrábí a je dokázáno, že na všechny překážky na silnici auto reaguje rychleji a lépe než sám člověk. Samozřejmě že člověk, který miluje řízení, tomuto vývoji moc nepodlehne a stále bude chtít mít vůz pevně v rukou. Však veliký zájem projevují dopravní společnosti, které nejvíce zatěžují právě chyby řidiče při dopravních nehodách anebo samotné platy zaměstnanců. V tomto případě by dopravce ušetřil za platy zaměstnanců, dokázal kontrolovat své kamiony z kanceláře a ještě stihl více zakázek, protože řidiči musí dodržovat přestávky. Při takto rychlém vývoji technologie, jaký zažíváme za posledních 20 let, se může brzy stát, že v běžném silničním provozu budou jezdit auta bez řidičů.



Technologií se zajímá celý svět, a proto vznikla taky soutěž NXP Cup, dříve Freescale Cup a Intelligent Car Race, v roce 2003 v Koreji. Od té doby se NXP Cup rozšířil do Číny, Indie, Malajsie, Latinské Ameriky, Severní Ameriky a Evropy. Tato soutěž je určena pro studenty středních a vysokých škol z celého světa. Pravidla pro tuto soutěž jsou striktně daná, například podvozek, který musí mít každý závodník stejný. Pro rok 2018 se vytvořil nový typ podvozku jménem Alamak. Starý podvozek Type-C se již nevyrábí, ale rovněž závodit může. Podle podvozků jsou rozděleny i kategorie soutěžících, aby tak nedocházelo k možným pochybnostem. Mozkem celého auta je programovací deska Freescale FRDM-KL25Z, kterou je taky nutno použít pro splnění podmínek. Auto musí urazit dráhu na náhodně sestavené trase v co nejkratším možném čase, a to bez vykolejení a zásahu obsluhy. Vozidlo se na trase pohybuje pomocí zpětné vazby z optické kamery a dalších možných senzorů, které zpracovává mikrokontrolér, a dále ovládá servomechanické řízení a dva motory poháněné stejnosměrným proudem. Je to třetí ročník kdy se bude VŠB účastnit této soutěže a je to jedna z podmínek pro splnění mé bakalářské práce. Ve své práci popisuji vše ohledně soutěže NXP a d vývoj, naprogramování a vylepšení modelu auta. [1]

1 Rešerše autonomního řízení modelu

Kapitola podává základní informace k autonomnímu řízení modelu, kde cílem je dodržet zmiňovaná pravidla a podmínky v této rešerši. Poukazuje na světové trendy v dané oblasti a jejich metody řízení.

1.1 Požadavky na model auta

Model musí být použitý ze soupravy NXP. Souprava zahrnuje podvozek vozidla, který lze upravit, ale musí zůstat ve stanovených rozměrech 250 mm (šířka) x 400 mm (délka) x 305 mm (výška) a vzdálenost kol se nesmí upravovat. Do podvozku je možnost vrtat otvory a montovat pomocné součásti. Lze upravit nebo odstranit pružiny a spojky. V případě použití kapotáže musí být vyjímatelná. Souprava dále obsahuje dva motorky na bázi stejnosměrného proudu a sadu pneumatik, ty se měnit dají, pokud však mají stejnou specifikaci a jsou schváleny společností NXP. Pro rok 2018 jsou schváleny dva modely aut, nový model Alamak a starý Type-C. Použit může být jen jeden procesor. Pro oba modely musí být použit mikrokontrolér Freescale FRDM-KL25Z. Model musí být poháněn pouze jednou baterií o kapacitě 7,2V a menší nebo roven 3000mAh, dobíjecí NiCd, NiMH nebo Li-ION. Není povolen zesilovač DC-DC, který by překročil napětí baterie a celková přidaná kapacita všech kondenzátorů nesmí překročit 2000 μ F. K navigaci auta musí být použit pouze optický senzor a dále může být přidáno až 16 senzorů NXP. [2]

| Parameters | LANDZO Alamak models | OLD Type-C models |
|----------------|---|---|
| Picture |  |  |
| Body structure | One-piece body | Segmented body |
| Size | 28.5×16×7cm | 28.5×16×8cm |
| Motors | 7.2v 380 motor×2 | 7.2v 260 motor×2 |
| Steering gear | 15kg. cm | 6.5kg. cm |
| Tire diameter | 65mm | 50mm |
| Wheel base | 16cm | 16cm |

Obrázek 1. Základní rozdíly mezi modely [2]

1.2 Pravidla soutěže

Účastníky soutěže mohou být pouze studenti ze středních nebo vysokých škol. Taktéž návrh a výroba auta musí být pouze prací studentů. Při závodech se očekává dobré sportovní chování a žádné podvádění, takovéto přestupky mohou vést k diskvalifikaci.

Před registrací se týmy musí rozhodnout zdali budou závodit na starém modelu Type-C, nebo zvolí nový model Alamak. Po registraci již není možné měnit model.

Před závodem je provedena inspekce vozidla odborným technikem, který kontroluje specifikace vozidla, rozměry a požadavky na model. Poté je autíčko odstaveno v oblasti kontroly, kde k němu nemá nikdo přístup, aby se tak zamezilo podvodným úpravám.

Týmy jsou vybrány v náhodném pořadí. Před startovací čarou musí jeden člen týmu rozhodčímu hlásit, že autíčko je připraveno k závodu do 2 minut od vyzvání. Poté co rozhodčí potvrdí, že je autíčko připraveno, musí opustit startovní oblast během 30 sekund. Každý tým má 3 pokusy na úspěšné dokončení trati. Pokud je potřeba odebrat auto ze závodní dráhy, může zakročit pouze jeden člen týmu a pak má každý tým 2 minuty na úpravu vozu. V případě úprav lze provádět konfiguraci parametrů přes palubní rozhraní, jako jsou spínače apod. Dále lze změnit úhel snímání kamery, vyměnit baterii a očistit kola. Žádné jiné úpravy nejsou možné.

NXP Cup je rozdělen na 3 kola, a to národní kvalifikační týmy, regionální finále a celosvětové finále. Každé 1. a 2. místo pokračuje do dalšího kola závodu. [2] [5]

1.3 Popis tratě

Trat' je v každém kole jiná a je odhalena až v den závodu. Trat' se skládá z několika částí, včetně devadesátistupňových rohů, přímých linek, šikan, kopců, křížených cest a startovní/cílovou čarou. Samotná trat' je vyrobena z ABS bílé plastové desky s jednopalcovou tlustou černou čarou na obou okrajích trasy. [2] [5]



Obrázek 2. *Příklad závodní dráhy [3]*

1.4 Analýza problematiky

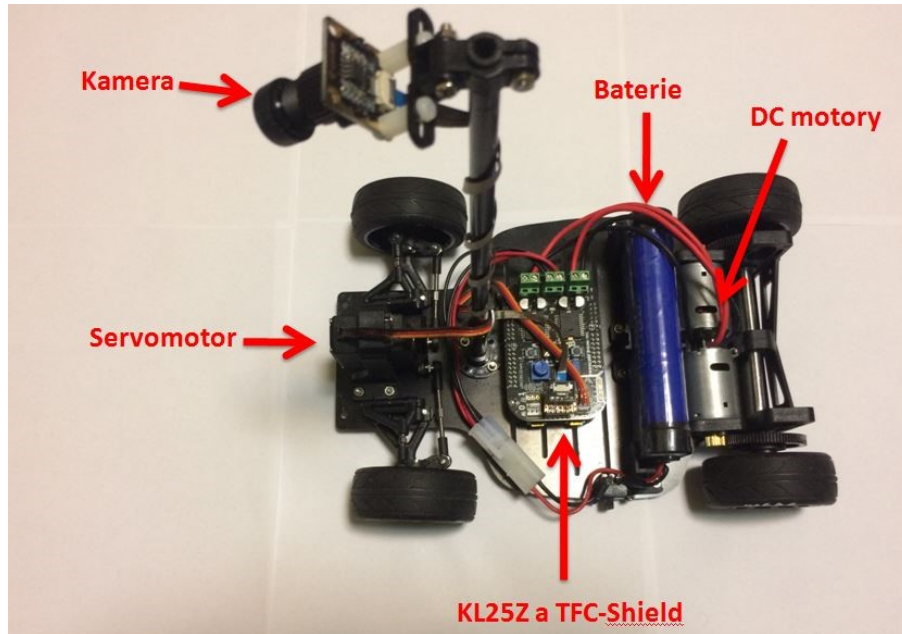
Z předešlých závodů lze usoudit, že důležité je sestavit co nejmenší program, který lze rychle přečíst, a komunikace mezi kamerou a mikrokontrolérem je okamžitá vůči změně trasy. Kamera jako taková sama nestačí, osvědčilo se již pár senzorů navíc, které se starají o lepší reakci na trati. Na modelu auta není žádná kapotáž, protože potřebujeme co nejnižší rychlost k dosažení většího výkonu auta. Při přejíždění přes horizont je nutné vyřešit rychlou stabilizaci kamery.

1.5 Světové trendy v autonomním řízení modelů aut

V dnešní době je již mnoho způsobů, jak řídit autonomně modely aut. V této práci je řízení pomocí detekovaného obrazu z kamery, to je nejčastější způsob řešení. Další způsob je pomocí ultrazvukového měření vzdálenosti, které je využíváno především u robotických vysavačů. Pokročilejším autonomním řízením je pomocí navigace se zadáváním trasy třeba přes mobilní aplikaci.

2 Návrh a realizace hardwarové části modelu

Základní hardwarová část modelu auta se skládá z desky řídicího systému Freescale FRDM-KL25Z a TFC-Shield, podvozku, 2x DC-morky, servomotoru pro řízení, baterie a kamery. Podrobnější specifikace pro tyto části jsou popsány v této kapitole.



Obrázek 3. Popis částí modelu auta

2.1 Freescale FRDM-KL25Z

Typ FRDM-KL25Z je vývojovou platformou a je základní jednotkou pro řízení vozidla. Tento freescale je vybaven mikrokontrolérem NXP KL25Z s maximální hodinovou frekvencí 48 MHz, vysoce výkonným jádrem ARM Cortex-M0+, pamětí programu 128 kB Flash paměti a pamětí dat 16 kB SRAM, což je vhodné pro zpracování dat z kamery a ovládání periferních desek s dostatečným množstvím úložiště a paměti. Má také rozhraní SPI, I2C a UART pro další připojení periférií. FRDM-KL25Z je první hardwarová platforma, obsahující NXP integrovaný sériový a debugovací adaptér s otevřeným standardem, známým jako OpenSDA. Tento typ se připojuje k počítači pomocí USB a konektor pro freescale je mikro USB. Je také vybaven RGB ledkou a snímačem kapacitního dotykového jezdce a trojosého akcelerometru. [2] [19]

2.2 TFC Shield

Tato deska je součástí standardní sady pro NXP Cup. Jejím základem jsou dva obvody MC33887. Jsou to dvojité H-můstky, které jsou zapojeny paralelně a každý motor tak může odebírat proud až 5 A. TFC nabízí další periferie, a to 4 LED diody, které lze naprogramovat pro kontrolu provedených funkcí, 2 potenciometry pro možnost nastavování rychlosti, 2 tlačítka využitá třeba na START/STOP, čtyřnásobný spínač pro různé režimy jízdy. Na TFC jsou konektory pro připojení řídicího serva, kamery a motorů. [2] [5]



Obrázek 4. Ukázka TFC-shieldu na Freescale desce [4]

2.3 Podvozek

Pro svůj model auta jsem zvolil novější typ podvozku Alamak. Oba typy podvozku od sebe nejsou zásadně odlišné, ale Type-C se již nevyrábí, a tak není dostupné ho koupit nový.

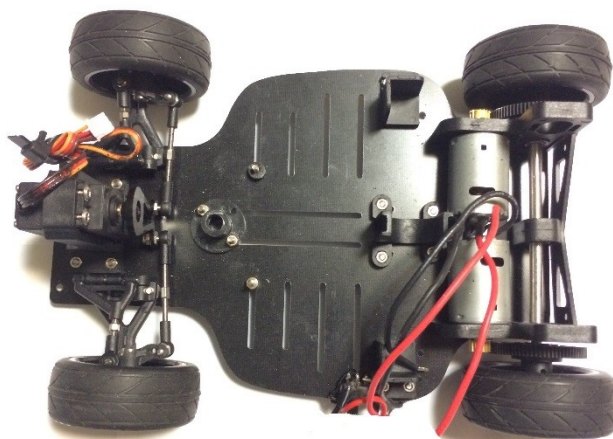
U podvozku se musí dbát na každý detail, protože to rozhoduje o výkonu vozu jak při projíždění zatáček, tak brzdění, zrychlení a udržení stability na trase. O zatáčení se stará servomechanické řízení předních kol, které se dá doladit k přesnějšímu ovládání, například změnou materiálu jednotlivých částí.

Pro správnou navigaci auta musí mít kamera dobré zorné pole, tedy záleží nejen na jejím úhlu naklonění, ale taky na správném umístění kamery na vozidle. Při umístění držáku s kamerou je třeba dbát na stabilitu celého modelu.

Důležitým faktorem jsou taky všechna kola modelu. Všechny kola musí mít stejnou tuhost a vzorek pneumatik. Při jakémkoli odlišném parametru na jednotlivých kolech může znamenat změnu ve výsledném čase několik sekund. Model má pohon zadních kol, proto je důležité, aby pneumatiky byly vždy čisté. Pokud kola nebudou splňovat ideální parametry a čistotu, může dojít v zatáčkách k nedotáčivosti či přetáčivosti kol. V těchto případech se může stát, že vozidlo se neudrží na závodní trati.

S umístěním baterie nelze manipulovat. Je umístěna na středu vozidla, a tak jsou dobře rozvrženy váhy. Avšak za uvážení stojí i fakt, že kdyby baterie byla umístěna více na zadní části, vahou působící na kola by mohla být přilnavost k dráze lepší.

Pro celkové zvýšení rychlosti je zapotřebí i dobré mazivo, které je vhodné jak pro plastové, tak kovové části. Mazání bude hlavně potřeba v okolí ozubených kol, tedy převody mezi motory a koly. Další částí bude taky servomechanické řízení, které díky mazivu získá menší tření součástí a hbitější zatáčení kol.



Obrázek 5. *Podvozek Alamak*

2.4 DC motory

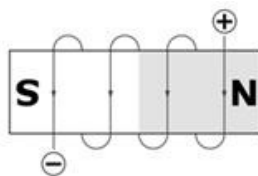
Model je poháněn dvěma motory na bázi stejnosměrného proudu, které jsou součástí standardní sady NXP Cup. Motory lze změnit, ale musí mít stejné specifikace a musí se řídit pravidly a předpisy NXP. Motory jsou na sobě nezávislé, takže je lze ovládat samostatně. To může pomoci při vyjíždění ze zatáček kdy jedno kolo se může otáčet pomaleji a druhé rychleji v závislosti na pravostranné či levostranné zatáčce. Pro každý motor je samostatná integrovaná převodovka umístěná vzadu vozidla a ta je připojena na jednotlivé zadní kolo.

Tabulka 1. Specifikace motoru [7]

| | |
|-------------------------|------------------------------|
| Označení | RN260-CN-18130 |
| Řídicí napětí | 7.2 V |
| Směr otáčení | CW pohled z kovového pouzdra |
| Poloha motoru | Horizontální |
| Provozní teplota | 10–30 °C |
| Provozní vlhkost | 30 % - 95 % RH |
| Bez zatížení | 16000 +/- 3200 ot. / min |
| Bez zatěžovacího proudu | 220 mA (max) |
| Mechanický hluk | 75 dB |
| Točivý moment | 7,8 x 10 ⁻³ Nm |

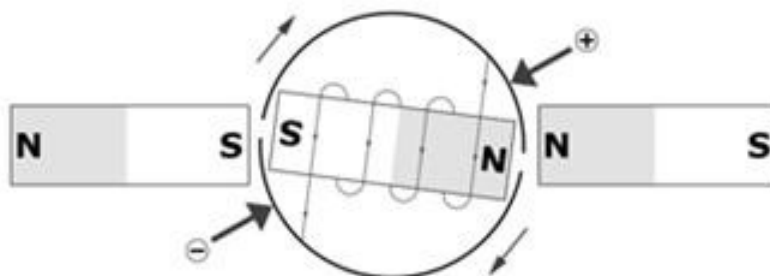
2.4.1 Princip DC motoru

DC motor přeměňuje elektrickou energii na mechanickou energii ve formě otáčení. Toto otáčení je způsobeno elektromagnetismem, který zajišťují induktory uvnitř motoru. Ty vytvářejí magnetické pole, které slouží ke generování pohybu.



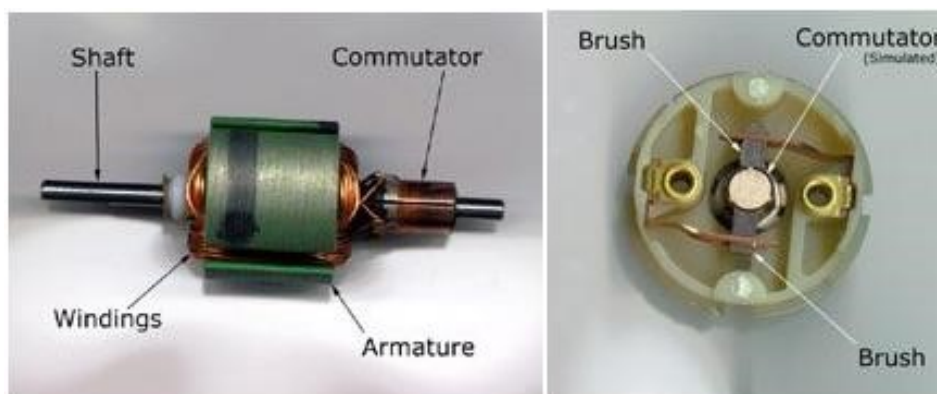
Obrázek 6. Elektromagnet [6]

Elektromagnet je kus železného tělesa obalený drátěnou cívkou, ta má na svých svorkách přivedeno napětí. Na dvou stranách elektromagnetu jsou přivedeny dva pevné magnety, tím že mezi nimi vzniká odpuzující a přitažlivé síly, se vytvoří točivý moment.



Obrázek 7. *Rotace elektromagnetu [6]*

Na hřídelce motorku je připevněn komutátor, který má dva segmenty připojené ke svorkám elektromagnetu. Jak můžete vidět na obrázku 1.7, jsou zde i takové svorky, na které je přiveden elektrický proud k rotačnímu elektromagnetu.



Obrázek 8. *Popis DC motoru [6]*

Po přivedení napětí na svorky se motor roztočí. Pokud chceme změnit směr otáčení hřídele, stačí zaměnit přivedené napětí na svorkách. Motorky mají tendenci se přehřívat, tudíž je potřeba dbát na využívání modelu. V tomto případě by mohlo být vynalezeno nějaké chladicí zařízení, ale pro naše využití modelu na závody by to mělo pouze negativní důsledky, protože auto by se zvýšila hmotnost. [6]

2.5 Servomotor

Servo je určeno k ovládání auta. Připojením k mikrokontroléru vzniká propojení, díky kterému je vyslán řídicí signál. Díky přivedenému signálu dochází k pootočení výstupní hřídele o požadovaný úhel. Čím přesnější bude toto otáčení, tím větší potenciál má model auta na rychlejší čas. Celková citlivost natočení je dána kvalitou řídicí elektroniky. Servo se skládá ze stejnosměrného motoru, převodovky a řídicí jednotky. [8]

2.5.1 Řídicí elektronika

Zpracovává vstupní signál, ze kterého určí úhel natočení, ten je porovnán s natočením zpětnovazebního potenciometru a pak až natáčí motor požadovaný směr. U klasických servomotorů je minimální úhel natočení o dva stupně, proto může vést i k menšímu zpomalení požadovaného času. U kvalitnějších lze pak dosáhnout minimálního natočení jeden stupeň. [8]

2.5.2 Servomotor Futaba S3010

Řízení modelu pomocí servomotoru Futaba je podle NXP standardního setu. Po zkouškách vytrvalosti a účinnosti servomotoru se ukázalo, že s plně nabitou baterií auto velice rychle reaguje. Při postupném vybíjení baterie na intenzitě zatáčení servomotor slábne, proto je potřeba mít plně nabitou baterii.

Tabulka 2. *Specifikace servomotoru Futaba [9]*

| | |
|-----------------|--|
| Rychlost | 0,20 s / 60 při 4,8 V; 0,16 s / 60 při 6 V |
| Točivý moment | 0,51Nm při 4,8 V; 0,64 Nm při 6 V |
| Napájecí napětí | 4,8 - 6 V |
| Rozměry | 40 x 20 x 38 mm |
| Hmotnost | 41 g |



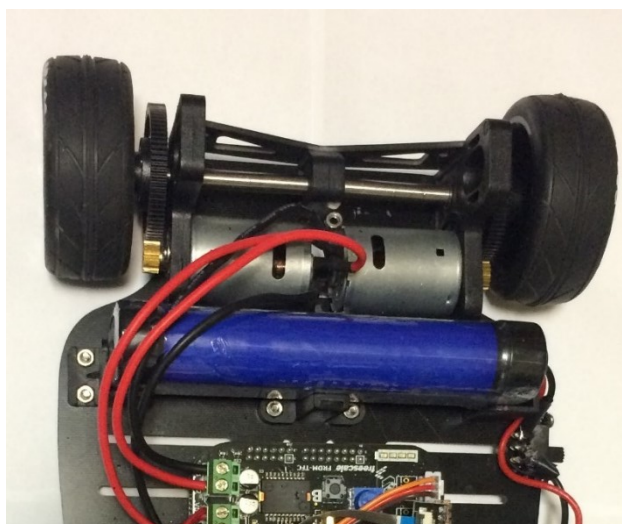
Obrázek 9. Servomotor Futaba S3010 [10]

2.6 Baterie

Volba baterie je omezená a její parametry musí být 7.2 V a ≤ 3000 mAh, typu NiCd, NiMH nebo Li-Ion. U výběru baterie je důležitá co nejnižší hmotnost a kapacita. Pro kratší tratě může mít i menší kapacitu, např. 1000 mAh. U takové baterie bychom pak určitě měli nižší hmotnost než u baterie o kapacitě 3000 mAh. Baterii s větší kapacitou bych využil při testování modelu a na samotné závody bych pak volil baterii s menší kapacitou. Baterie byla dodána k podvozku s kapacitou 2700 mAh, ta byla použita při testování i závodění.

Tabulka 3. Srovnání baterii [11] [12]

| Výrobce | Kapacita[mAh] | Napětí[V] | Hmotnost[g] |
|---------|---------------|-----------|-------------|
| Nosram | 3000 | 7,2 | 260 |
| Traxxas | 1200 | 7,2 | 144 |



Obrázek 10. *Umístění baterie na podvozku*

2.7 Kamera

Jedná se o řádkovou černobílou kameru, která je osazena čipem TSL1401CL. Výstup z kamery je analogový, s rozlišením 128 pixelů. Na desce plošného spoje má kamera vlastní snímací čip, neinvertující operační zesilovač a optiku. Digitalizace analogového výstupního signálu je vyřešena pomocí A/D převodníku v mikrokontroléru. [5]

Pro přesné snímání kamery je třeba zmírnit otřesy, aby kamera snímala přesně dráhu a nedocházelo k jejím výpadkům. I když na závodní trati nejsou žádné kameny apod., stále se autíčko dostává do mírných otřesů, a to například při projetí zatáčkou, anebo přes horizont, který na závodní trati taky často je. Řešíme tedy stabilitu držáku i samotné kamery. Základním bodem je správně umístit kameru na takové místo, kde bude pro kameru dobré zorné pole s minimální možností nárazu při vykolejení z dráhy.

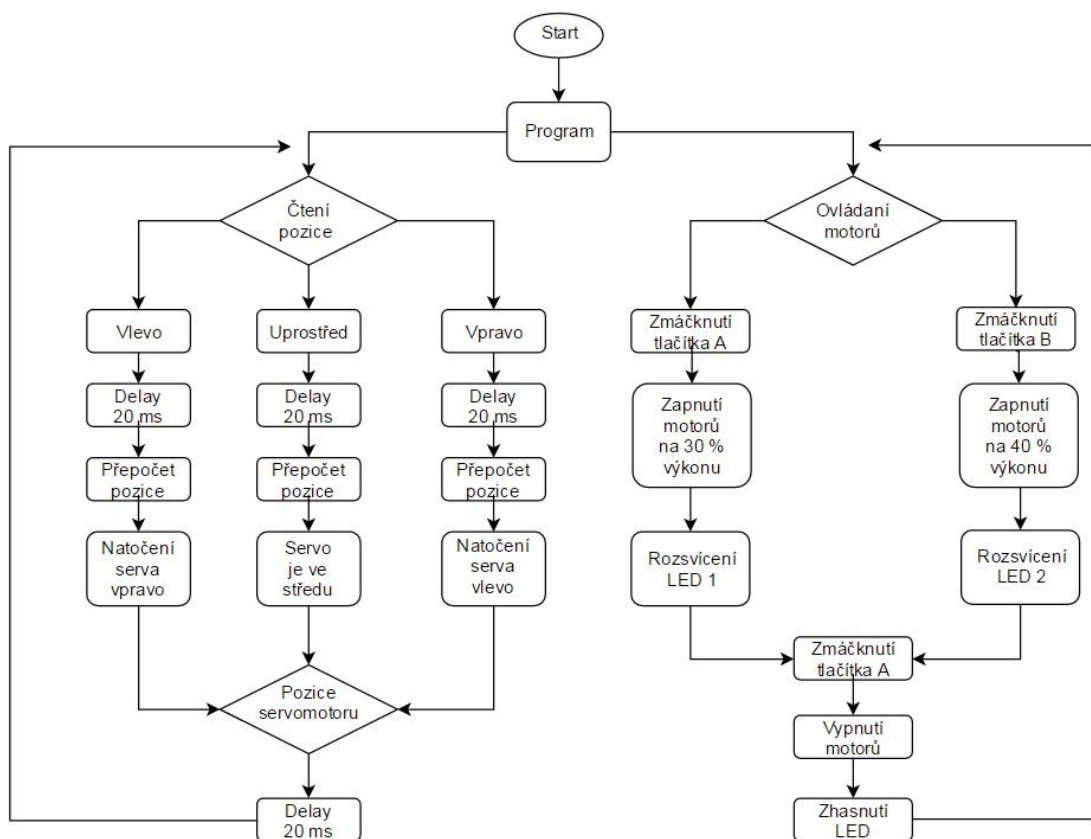
Upevnění kamery je na hliníkové tyči a pro její upevnění na podvozku je třeba vymyslet uchycení takové, aby nebylo příliš tuhé. Nejideálnějším řešením by byl stativ přímo na držáku kamery, který by stále udržoval stejné zorné pole i při přejíždění horizontu. Dalším vylepšením mohou být přidána osvětlení na tyči tak, aby kamera měla lepší snímání dráhy.



Obrázek 11. *Použitá kamera na modelu auta*

3 Návrh a implementace řídicího softwaru

Řídicí software byl implementován pomocí programovacího prostředí Atolic TrueStudio, které umožňuje programování pomocí jazyka C. Program byl zvolen, protože je volně ke stažení a je kompatibilní pro procesor ARM Cortex-M0 na řídicí desce. Na vozidle je použito mnoho komponentů, které je třeba použít pro optimální autonomní řízení. Proto bylo zapotřebí vytvořit strukturu kódů, která je srozumitelná a dá se s ní efektivně pracovat. Pro každou funkci je vytvořen zdrojový soubor se zaměřením na danou úlohu. Pro tyto zdrojové kódy jsou vytvořeny i hlavičkové soubory, které se volají v kódu v případě jejich využití. [19]



Obrázek 12. Vývojový diagram programu

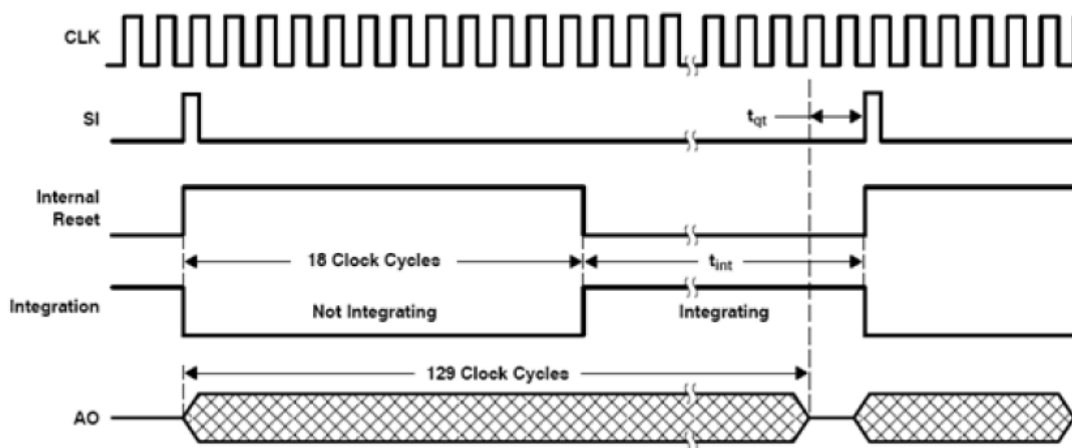
3.1 Analogově-digitální převodník

Pro přenos signálu z kamery je zapotřebí ADC převodník. Jeho inicializace je ve zdrojovém souboru `adc.c` a nastavena ve funkci `adc_init(void)`. Nejprve jsou ve funkci povoleny hodiny pro analogově-digitální převodník 0. Následně je vybrán konfigurační registr CFG1, který vybírá režim provozu, zdroj hodin, dobu vzorkování. Registru je přiřazen ADIV(3) tedy poměr dělení 8 pro generování interních hodin ADCK. Dále je nastaven ADC MUX na konfiguračním registru CFG2. Ve funkci `adc_read(channel)` probíhá čtení analogové hodnoty a její hodnota se zapisuje na kanál 0.

3.2 Detekce obrazu

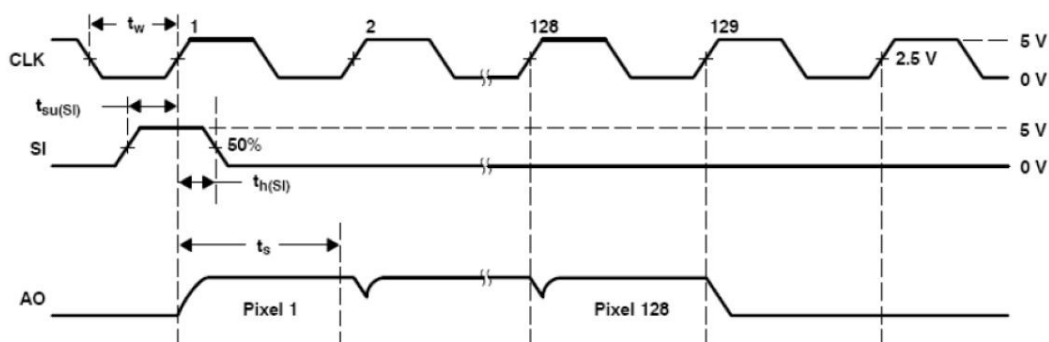
Funkce pro vyčítání z kamery je obsažena ve zdrojovém kódu `camera.c` s inkludovaným souborem `camera.h` a ve funkci `camera_init(void)` se provádí inicializace. Povolení hodin je pro port D pro pin 5, 7 a port E pro pin 1. Ke každému pinu je určen řídicí multiplexor a u pinu 5 je nastaven na analogově-digitální převodník, pro pin 7 a 1 je alternativa GPIO, tedy slouží jako univerzální vstup a výstup. Dále je nastaven PDDR na výstup pro port E pin 1 a pro port D pin 7.

Pro čtení z kamery jsou použity tři signály a to CLK, SI, AO. Frekvence snímání signálů z kamery je volitelná uživatelem. Při použití vysoké rychlosti čtení může dojít k horší kvalitě pixelů z důvodů jejich nenabití a, proto je důležitá doba integrace pixelů (viz. obrázek 13). Doba integrace je čas, který pixely musí dokončit. Při dlouhém integračním čase budou pixely nasyceny, i když je nízká intenzita světla v prostředí. Naopak když bude velmi krátký integrační čas, pixel se nenabije, i když je nadměrné světlo v prostředí. Jak je vidět na obrázku 13, po 18hodinových cyklech se pixely začnou znovu nabíjet a následně po 128hodinových cyklech jsou všechny pixely uvolněny z kamery. V tento moment můžeme znovu poslat start impuls pro další uvolnění výstupního signálu. Mnou zvolená rychlost čtení z kamery je shodná s periodou PWM při ovládání servomotoru, a to tedy 20 ms. [13]



Obrázek 13. Integrační čas [13]

Všechny tři signály (CK, SI, AO) jsou synchronizovány (obrázek 14). Pro správné vyčítání z kamery musí signály SI a CLK mít stejnou šířku impulsu. Oba signály musí být dokonale zarovnány pro polovinu pulzu. Za těchto podmínek bude AO sladěn s CLK. [13]



Obrázek 14. Synchronizace signálu SI a CLK pro získání AO [13]

Ve zdrojovém kódu `camera.c` tedy definuji funkce pro nastavení logické 1 a 0 na daném signálu, a to konkrétně `set_SI` s příkazem `PSOR` na pin 7, tak stejně funkci `reset_SI` s příkazem `PCOR` na pin 7. Další funkce jsou `set_CLK` a `reset_CLK` na pin 1. Tyto funkce jsou později použity v kódu pro spuštění start impulsu a hodinového signálu.

Funkce `read_camera(void)` se stará o vyčítání z kamery kde jsou použity výše zmiňované signály. Každá šířka pulzu je nastavena na 20 μs . Základem je start impuls `set_SI` a v polovině tohoto pulzu tedy po 10 μs se spouští hodiny `set_CLK`. Po dalších 10 μs se resetuje start impuls `reset_SI` a po 10 μs se čte analogová hodnota pomocí `adc_read` z kanálu 6 a tuto hodnotu zapisujeme do pole pixelů `cam_array` od nulté hodnoty po 128. Následně jsou resetovány hodiny `reset_CLK`. Důležité nastavení je, aby se CLK vypnul po načtení 128 hodnot. Proto ve smyčce `for` je napsáno, dokdy má kamera pole zapisovat a vypnout CLK.

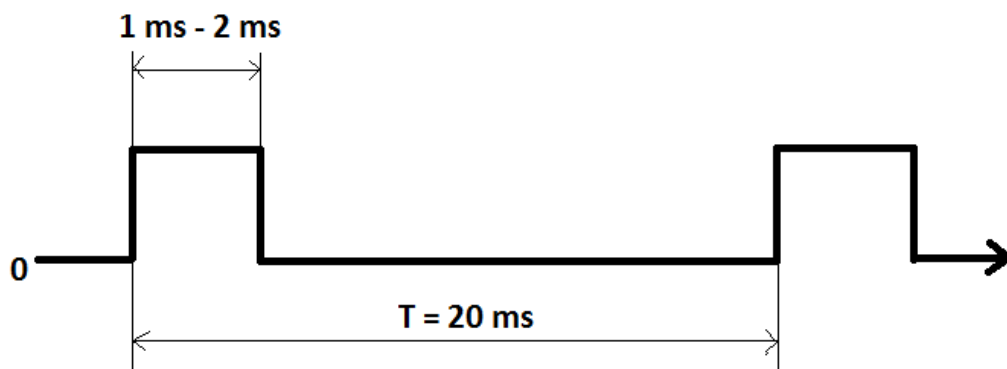
3.3 Ovládání řízení

Ve zdrojovém kódu `servomotor.c` je nejprve inkludovány jeho hlavičkový soubor a dále definovaná makra pulzu servomotoru. Jeho minimální hodnota zatočení je nastavena na 1000(zatočení vlevo) a jako maximální hodnota 2000(zatočení vpravo). Inicializace je provedena s názvem `servo_init(void)`. Zde jsou povoleny hodiny pro port B pro pin 0 a MUX 3. Dále je provedeno povolení hodin pro časovač 1 s frekvencí 8 MHz. Jako zdroj hodin je SOPT2 a zde je přiřazen čítač TPM s povolením hodinového signálu OSCERCLK, což odpovídá frekvenci 8 MHz. Následuje nastavení předděličky, kde pracujeme s 8 MHz, a požadovaná frekvence je 25 kHz, zvolil jsem tedy vhodnou předděličku (3) a MOD vyšel na hodnotu 20000-1. Pokračuji nastavením kanálu 0 a jemu přiřazení druhu PWM a následně přiřazená hodnota PWM na 1500 aby servo při zapnutí bylo ve středu. Následně už jen povolení časovače jeho výstupu.

Do funkce `servo_set (uint16_t length)` zadávám šířku pulzu a ta je přepsána na výstup kanálu 0. Pokud spadá šířka pulzu do makra `MIN_PULSE`, zapíše její hodnotu na výstup a pokud spadá šířka pulzu do makra `MAX_PULSE` zapíše hodnotu na výstup.

3.3.1 Zpracování signálu PWM

Pulzně šířková modulace je diskretní pro přenos analogového signálu pomocí dvoustavového signálu vypnuto/zapnuto, tedy střída, která má určitou periodu. U servomotoru je perioda 20 ms při frekvenci 50 Hz. Pro otáčení servomotoru je důležitá šířka daného pulzu. Jak můžete vidět na obrázku 1.14, šířka pulzu je v rozmezí 1 ms - 2 ms. Střed servomotoru je tedy při šířce pulzu 1,5 ms a pro otočení doleva bude mít pulz šířku menší než 1,5 ms anebo rovno 1 ms. Pro otočení doprava bude mít pulz šířku větší než 1,5 ms anebo rovno 2 ms.



Obrázek 15. PWM cyklus pro servomotor

3.4 Detekce polohy

Autíčko se snaží zůstat vždy ve středu trati. Jedná se o jednoduchý a velmi spolehlivý přepoččet pozice. Tato funkce funguje pomocí kamery, která detekuje posunutí vozidla od středu trati a tím řídí servomotor. Aby kamera správně snímala, musí být řádně upevněna na podvozku a musí být minimalizovány odrazy a stíny na trati. Pomocí měření v Atolic Studiu jsem kalibroval zaostření kamery a ohnisko vzdálenosti tak, aby bylo autíčko schopno včas zatočit při sledované zatáčce.

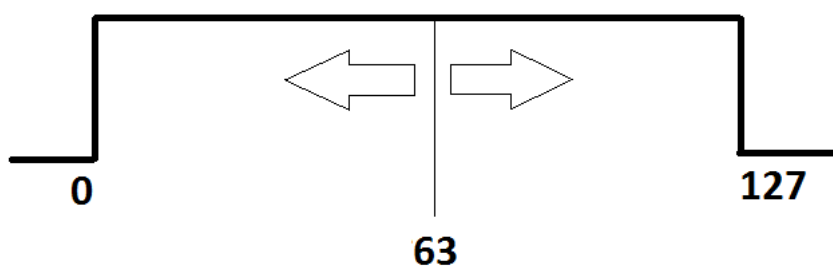
3.4.1 Popis řešení v programu

Pro zjištění polohy autíčka na trati jsem ve funkci `position_set(void)` definoval parametry s maximální a minimální hodnotou, kde je černá hranice, určil polohu auta vlevo `startL = 0`, určil polohu vpravo `startP=127`, proměnnou `position`, `i`, `avg`.

Voláním `read_camera()` se funkce spustí a následně je vykonávána smyčka `for`, která čte hodnoty z pole pixelů, toto pole je však omezené od 10 do 118, aby došlo k včasnému zatočení, v případě snímání krajních hodnot. Pokud je pole větší než maximální hodnota, zapíše se její hodnota do `max`. A naopak pro minimum. Jestliže minimum je menší než hodnota 230, provede se průměr sečtení hodnot `max` a `min`. V další smyčce `for` je proměnná `i` rovna středu dráhy, tedy číslu 63. Průměr se porovnává ve smyčce `for`, kde zjišťujeme, jestli pole hodnot je menší než průměr a platí pozice vlevo `startL`, pokud ano, zapíše se hodnota z pole do pozice `startL`. Jestli je průměr větší než rozdíl celého pole s proměnnou `i` a platí pozice vlevo `startR`, je hodnota `127-i` zapsána do pozice `startR`. Pak finální pozice je vypočtena pomocí vzorce:

$$position = 128 - startL - startR$$

Pokud neplatí ani jedna podmínka pak `position = 0` a autíčko je tedy ve středu.



Obrázek 16. *Určení středu pozice na dráze*

3.5 Nastavení motorů

Na začátku zdrojového souboru `motors.c` je inkludovaný hlavičkový soubor `motors.h`. Poté je funkce `motors_init(void)`, kde je provedena inicializace, a to povolení hodin pro port C a E. Všechny porty C pinu 1,2,3,4 jsou nastaveny na MUX (4) a port E pinu 21 nastaven na MUX(1). Dále je nastaven PDDR na výstup pro port E pin 21 a PSOR pro port E pin 21. Povolení hodin pro časovač 0 a zdroj hodin je stejný jako u servomotoru. Nastavení předděličky s frekvencí 8MHz a požadovaná frekvence je 25 kHz, zvolil jsem tedy vhodnou před děličku (3) a pomocí výpočtu, MOD vyšel na hodnotu 101-1. V tomto případě je možno volit délku střídání u PWM od 0 do 100, nastavení kanálu 0,1,2,3 na hodnotu 0 pro daný časovač. Následně pokračuje povolení časovače a jeho výstupu.

Kód pro oba motory je ve funkci `motors_set(left, right)`. U těchto parametrů je uváděn i datový typ `uint16_t`. Funkce popisuje meze pro motory. Pokud je pravý motor nastaven na hodnotu menší než 1, je vypnutý. Pokud je hodnota větší jak 100, je rovna 100. Na konci programu je povolení hodnoty pro pravý motor na kanálu 0 a levý motor na kanálu 2. Tyto kanály jsou využity pro pohyb dopředu. Později bych navrhoval pohyb dozadu jako možnost brzdění a v tomto případě by byl využit kanál 1 a 3.

3.6 Nastavení tlačítek

Na začátku zdrojového souboru `buttons.c` je inkludovaný hlavičkový soubor `buttons.h`. Inicializace tlačítek je provedena ve funkci `buttons_init(void)`, povolení hodin pro port C a následně nastavení MUX(1) pro pin 13 a 17. Vytvořeny jsou dále funkce `buttons_read` pro tlačítka A i B. Jestliže se na Port Data Input Register bude přivedena hodnota z pinu 13 a nebude rovna nule, vrátí se na výstup 1 pro možné čtení hodnoty. Tak stejně tomu je u tlačítka B, ale na pinu 17.

3.7 Nastavení LED

Na začátku zdrojového souboru `leds.c` je inkludovaný hlavičkový soubor `leds.h`. Inicializace je provedena ve funkci `leds_init(void)`, povolení pro port B a nastavení MUX(1) pro piny 8,9,10,11, nastavení pinů na výstup Port Data Direction Register. Ve funkci `leds_set(led, state)` je použit přepínač switch s argumentem v závorkách (`led`). Na základě tohoto argumentu přeskočí program na návěští (`case`) se stejnou hodnotou a pokračuje vykonáváním příkazu za ním. V každém návěští je nastaven PSOR a PCOR na pin dané led.

3.8 Hlavní smyčka

V hlavní funkci `main(void)` datového typu `int` jsou inkudovány všechny potřebné zdrojové soubory, které jsou potřebné ve smyčce `while(1)`. Jestliže je zmáčknuto tlačítko A, motory jsou sepnuty na 30 % svého výkonu. Jako zpětnou vazbu, že byla funkce provedena je rozsvícení jedné LED diody na TFC-Shieldu. Pokud je tlačítko A stisknuto znovu, motory se vypnou a LED dioda taky. Druhé tlačítko B je využito taky na motory ale s výkonem na 40 % a zpětnou vazbou je rozsvícení druhé LED diody. Následně je volána funkce `read_camera()` a po 20 ms se začne vyčítat pozice serva. Proměnná `tmp` je rovna vypočtené polohy auta na dráze a následně je důležitý přepočet pro natočení serva pomocí vzorce:

$$tmp = 1500 + (3 * tmp)$$

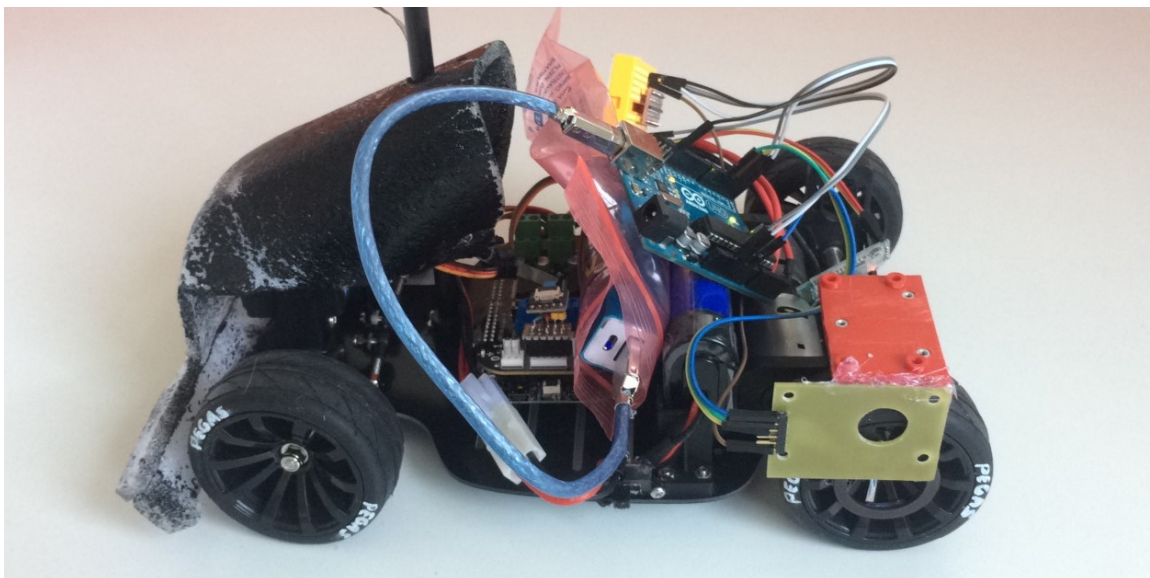
Hodnota 1500 je střed serva a k němu musíme připočíst pozici na dráze vyčtenou z kamery. Pokud tedy bude hodnota `tmp` kladná, servo se otočí doprava, pokud záporná, otočí se doleva, a to pomocí funkce `servo_set(tmp)`. Takto probíhá jedna smyčka za druhou pořád dokola.

4 Návrh, realizace a porovnání regulačních algoritmů pro optimalizaci provozu modelu.

Kapitola zahrnuje možné optimalizace pro provoz vozidla. Pro takové optimalizování a zjištění potřebných úprav je zapotřebí provést měření různých faktorů ovlivňující provoz vozidla. Díky těmto faktorům můžeme dále eliminovat nežádoucí účinky pro provoz.

4.1 Identifikace motorů

Pro měření motorů byla použita platforma mikrokontroléru ARDUINO se zapojením magnetického snímače polohy AS5306 v kombinaci s magnetickým kroužkem MR12-72, který byl připevněn na zadní kolo autíčka. [15] [16]



Obrázek 17. *Zapojení pro identifikaci motorů*

4.1.1 Popis programu v Arduinu

Program pro měření obsahuje funkci `setup` pro inicializaci a funkci `loop` jako nekonečnou smyčku. Na výstupu jsou dva kanály A, B a na nich jsou generovány pulzy podle toho, jak se magnet otáčí. Tyto kanály jsou oproti sobě posunuté v signálu, takže je lehce znatelný směr otáčení. Dále jsou definované piny, které jsou připojeny na kanály. Na jeden z nich je v programu aktivováno přerušování detekující hrany na signále A z kvadrurního inkrementálního enkodéru. Při detekci hrany proběhne podle hodnoty na kanále B inkrementace nebo dekrementace čítací

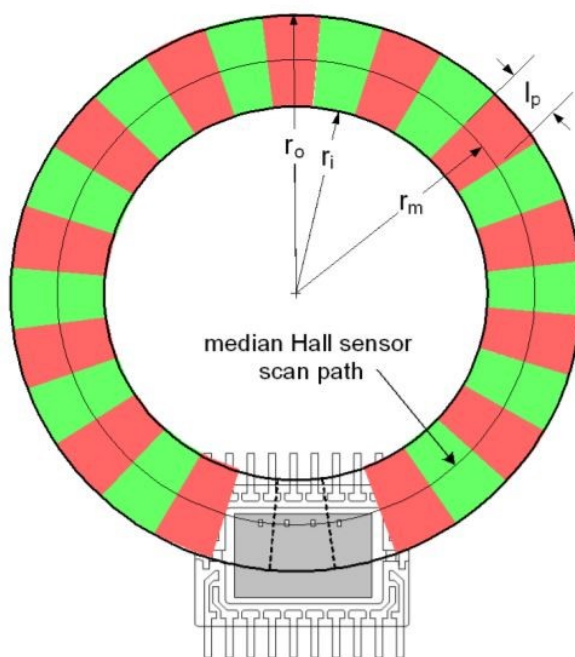
proměnné. Dále je aktivní přerušení od timeru co 5 ms, ve kterém se z rozdílu aktuálního a posledního počtu načítaných hran určuje rychlost otáčení motoru. V části `setup` je zapnut motor na hodnotu plnění PWM signálu 40 z maximálních 255 (vytvoření skoku) a co 5 ms je tato hodnota odesílána přes virtuální sériový port. K zobrazení byl využit sériový monitor, kde jsou naměřené hodnoty číselně. Získané hodnoty jsou dále používány v Matlabu. [14]

4.1.2 Vlastnosti magnetického snímače a kroužku

Pro správné měření musí být středová osa magnetického prstence přes pole Hallového senzoru, který je 4,22 mm nad okrajem prvního pinu vlevo dole. Dalším faktorem pro správné měření je vzdálenost mezi senzorem a kroužkem, zde je doporučena vzdálenost do 1 mm. Magnetický kroužek je navržen tak, aby délka pruhu l_p odpovídala požadované délce při skenování. Správný poloměr měření lze vypočítat z počtu pólů magnetického kroužku a zadané délky jednoho pólu. [15] [16]

$$r_m = \frac{\text{pole_length(mm)} * \text{number_of_poles}}{2 * \pi}$$

$$r_m = \frac{1.2 * 72}{2 * \pi} = 13.75 \text{ mm}$$



Obrázek 18. Správné umístění prstence [16]

AS5306 je přírůstkový snímač polohy pro rotační snímání na bázi bezdotykové technologie magnetických čidel. K dispozici je 160 kroků na 2,4 mm délky pólového páru s maximální rychlostí 15 m/s. Pro měření je potřeba použít vícepólový magnetický kroužek a ten má délku jednoho pólu 1,2 mm a celkový počet pólů na kroužku je 72. Z těchto informací je výpočet rozlišení dán vzorcem. [15] [16]

$$\text{Rozlišení} = \text{počet kroků} * \text{počet pólů}$$

$$\text{Rozlišení} = 160 * 36 = 5760 \text{ kroků za otáčku}$$

Maximální lineární rychlost jízdy je dána vzorcem:

$$\text{Maximální lineární rychlost} = \text{maximální počet párů za sekundu} * \text{délka dvojice pólů}$$

$$\text{Maximální lineární rychlost} = 5000 * 2,4 = 12 \text{ m/s}$$

4.1.3 Výpočet naměřené maximální lineární rychlosti

Motory při měření byly zapnuty na 40 % výkonu. Naměřené hodnoty Arduinem byly zapisovány pomocí Bluetooth na počítač. Z těchto hodnot lze vyhodnotit počet párů za sekundu. Za 3 sekundy byl počet pulzů 600. Za 1 sekundu je tedy 200 pulzů. Jeden pulz = dva páry, a proto platí:

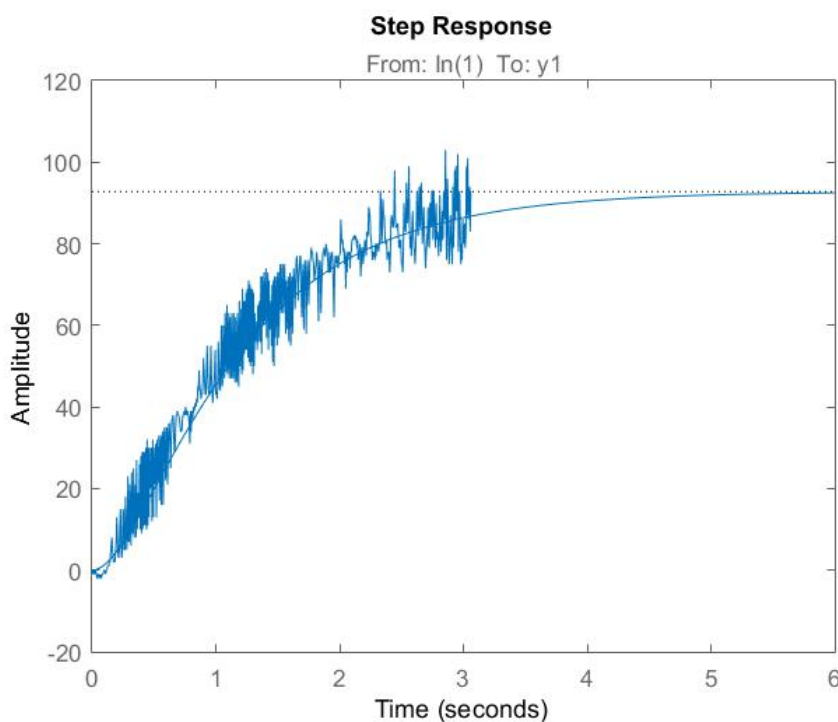
$$\text{Počet párů za sekundu} = 200 * 2 = 400$$

Takže pro maximální lineární rychlost měření při výkonu motorů na 40 % platí:

$$\text{Maximální lineární rychlost} = 400 * 2,4 = 0,96 \frac{\text{m}}{\text{s}} = 3,456 \frac{\text{km}}{\text{h}}$$

4.1.4 Popis programu v Matlabu

Identifikace motoru byla provedena pomocí programu Matlab. Nejprve byla založena proměnná output, do ní byly vloženy naměřené hodnoty. Další proměnnou je perioda vzorkování $T = 0.005$ s. Následně byl vytvořen signál odpovídající vstupu do soustavy motoru, tedy vektor hodnot 40. Poté byl použit nástroj pro identifikaci soustav příkazem `ident` a do něj byly importovány data ze vstupního a naměřeného signálu soustavy. Pro naměřená data byl zvolen obecný předpis přenosu se dvěma póly a bez nul a dopravního zpoždění. Nakonec byl vytvořen přenos soustavy a vykreslení přechodové charakteristiky identifikovaného přenosu a naměřených dat. [14] [18]




Obrázek 19. Přechodová charakteristika identifikovaného přenosu a naměřených dat

5 Test modelu na reálné trati

Model auta bylo potřeba po celý čas vyvíjení testovat na trati. Počínaje kamerou, která musí správně vyčítat obraz a určit tak, kde je střed trati a kde se blíží k černé čáře, a to vpravo, nebo vlevo. Dalším ovlivňujícím faktorem je poloha kamery. Kameru jsem tedy nastavil na dostatečné zorné pole tak, aby auto stihlo na trati včas zatočit v případě blížící se zatáčky. Zatáčku je však možno projet pouze s přesnou regulací servomotoru. Po celou dráhu se autíčko snaží udržet ve středu dráhy, díky přesnému čtení z kamery a regulací servomotoru dokáže auto jet rovně bez zbytečných výkyvů.

5.1 Soutěžní kolo NXP Cup

Součástí práce byla účast na národním kole soutěže NXP Cup, které se konalo 20.3.2018 na naší škole v Ostravě, konkrétně v budově FEI. Soutěže se účastnily týmy ze středních a vysokých škol. V kategorii s podvozkem Alamak se zúčastnilo celkem 9 týmů. Na soutěž jsem měl připravené dva rychlostní režimy, které se daly změnit pomocí tlačítek na TFC-Shieldu. Před závodem probíhal trénink na dvou dráhách pro konfiguraci vozidel na danou dráhu. Finální závodní dráha byla sestavena jinak než tréninková a nebylo ji možno odzkoušet. Proto jsem zvolil jeden režim rychlejší a druhý pomalejší. Po prvním pokusu, kdy autíčko vyjelo z dráhy, jsem zvolil pomalejší rychlostní režim a s časem 29,9 sekundy jsem se umístil na 3. místě. Ostatní týmy nemají zaznamenaný čas, protože jejich modely do cíle nedojely.



| University | Country | Team Name | Ranking | Round 1 | Round 2 | Round 3 | Lap Time | Comments |
|---|----------------|------------------------|---------|---------|---------|---------|----------|----------|
| Wroclaw University of Technology | Poland | UNI - Synergy | 1 | 00:17,2 | | | 00:17,2 | |
| Wroclaw University of Technology | Poland | UNI - Synergia | 2 | 00:20,8 | | | 00:20,8 | |
| Technical University of Ostrava | Czech Republic | UNI - The Pegas | 3 | 00:29,9 | | | 00:29,9 | |
| Technical University of Ostrava | Czech Republic | Blind Driver | DNF | | | | | |
| Technical University of Ostrava | Czech Republic | no piston | DNF | | | | | |
| Technical University of Ostrava | Czech Republic | UNI - CAREl | DNF | | | | | |
| AGH University of Science and Technology Krakow | Poland | UNI - KNE::Fideltronik | DNF | | | | | |
| Slovak University of Technology in Bratislava | Czech Republic | UNI-ELITA | DNF | | | | | |
| Technical University of Ostrava | Czech Republic | UNI - Amblinc | DNR | | | | | |

Obrázek 20. Kvalifikační tabulka v kategorii Alamak [17]

6 Zhodnocení práce a závěr

Cílem této práce byl návrh a implementace autonomního řízení modelu auta s použitím modelu a komponent využitých v soutěži NXP Cup. Nejprve jsem se seznámil s pravidly této soutěže a prostudoval doporučenou literaturu, abych porozuměl systému autonomního řízení. Jako vývojové prostředí jsem zvolil Atolic TrueSTUDIO, ve kterém jsem se naučil programovat desku Freescale FRDM-KL25Z. S připojením desky na TFC-Shield jsem začal pracovat s ovládáním servomotoru a dalších komponent. Pro správné zatáčení jsem mechanicky upravil servomotor na střed a v programu určil jeho maximální natočení vpravo a vlevo. Následně jsem prostudoval manuál ke kameře a naprogramoval čtení hodnot z kamery. Stěžejním úkolem bylo správně vyčítat údaje z kamery a následně vytvořit správný propočtení, o jakou hodnotu se má servomotor otočit v případě, že není ve středu dráhy. Díky testovací dráze, která byla k dispozici ve škole, jsem mohl tuto problematiku vyladit ke zdárnému výsledku. Model auta byl poháněn DC motorky, které jsou na každém kole zvlášť. Můj program nevyužívá možnost regulace motorů a jsou nastaveny na konstantní hodnotu výkonu.

Vzhledem k tomu, že autíčko bylo schopno dokončit závod v národním kole a umístit se tak na 3. místě, i když bylo dost konkurenčních uchazečů, považuji umístění za velký úspěch v této soutěži. Skutečnost, že jsem se takto kvalifikoval s využitím pouze 30% výkonu motoru, dokazuje, že autíčko má potenciál být ještě rychlejší. Zde bych navrhnul regulaci motorů pro využití maximální rychlosti při dlouhé rovině a v případě projíždění zatáček zpomalení či regulaci obou motorů s jinou rychlostí. Dalším vylepšením pro příští kvalifikaci bych navrhoval výměnu kvalitnějšího servomotoru, protože při jízdě má nežádoucí výkyvy, které ovlivňují výsledný čas. Výkyvy jsou způsobené vůlí servomotoru, ale taky mechanickým zpracováním hřídele ke kolům. I zde vznikají nežádoucí vůle a pro jejich eliminaci bych navrhoval pevné spojení obou tyčí se servem. Po konzultaci s konstruktérem jsem zjistil, že důležitým faktorem je taky sbíhavost a rozbíhavost kol. Při dodání podvozku měly kola velkou sbíhavost, a tak docházelo při jízdě ke smyku předních kol, což výrazně zpomaluje autíčko. Kola jsem seřídil na ramenech a změnil délky spojovacích tyčí stejnsměrně na obou stranách.

K novému podvozku Alamac byly vytvořeny i řídicí desky, které se lišily od starých svým designem a především spolehlivostí. Bohužel tyto desky byly velmi otevřeny destrukci a vznikaly u nich nežádoucí zkratky, odpalovaly se vypínače, kamera a taky PWM pin pro motor. Proto jsem později zvolil řídicí desky ze starého podvozku Model-C, které tento problém neměly a tato záměna desek je tolerována i v podmínkách NXP Cup.

Při procesu navrhování jsem se mnohému naučil, a to hlavně oblasti vyčítání dat z kamery, následně jejich zpracování pomocí ADC převodníku a komunikace se servomotorem. Vzhledem k účasti na soutěži NXP Cup, mohu konstatovat, že o soutěž se zajímá mnoho studentů, a to z různých zemí. I díky této soutěži se může kdokoli dále zabývat autonomním řízením aut a podpořit tak vývoj technologie.

Použitá literatura

- [1] KANTORKOVÁ, Táňa. Na VŠB – Technické univerzitě Ostrava budou soutěžit roboti!. Novinky: Tiskové zprávy [online]. 2016, 2016, 1 [cit. 2018-02-11]. Dostupné z: <https://am.vsb.cz/info/?reportId=30480&lang=cs>
- [2] STIFFAN, Flavio. The NXP Cup Official Rules: Season 2017/18. *NXP Community* [online]. 2017, 2017 [cit. 2018-04-29]. Dostupné z: <https://community.nxp.com/docs/DOC-335083>
- [3] JANALÍK, Stanislav. Ostrava zažila velkou premiéru. Vysoká škola báňská hostila NXP Cup. *Český rozhlas* [online]. 2017, 5. duben 2017, 2017 [cit. 2018-04-05]. Dostupné z: <https://ostrava.rozhlas.cz/ostrava-zazila-velkou-premieru-vysoka-skola-banska-hostila-nxp-cup-6953128>
- [4] *Connect TFC Shield to the NXP FRDM-KL25Z Board* [online]. [cit. 2018-04-29]. Dostupné z: <https://www.mathworks.com/help/supportpkg/freedomboard/ug/connect-tfc-shield-to-the-frdm-kl25z-board.html>
- [5] KRÁL, Jiří. NXP Cup (1). Robodoupě: web nejen o robotice [online]. 2016, 2016, 1 [cit. 2018-04-25]. Dostupné z: <http://robodoupe.cz/2016/nxp-cup-1/>
- [6] XIAOLI, GUO. How does a DC Motor work?. *NXP Community* [online]. 2012, 2012(1067), 1 [cit. 2018-03-25]. Dostupné z: <https://community.nxp.com/docs/DOC-1067>
- [7] *Freescale Cup DC Motor Specifications* [online]. 2012 [cit. 2018-04-05]. Dostupné z: <https://community.nxp.com/docs/DOC-93309>
- [8] Modelářské servo. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2018-04-29]. Dostupné z: https://cs.wikipedia.org/wiki/Model%C3%A1%C5%99sk%C3%A9_servo
- [9] *Futaba S3010* [online]. [cit. 2018-04-29]. Dostupné z: <https://www.peckamodel.cz/produkt/rc-modely-a-prislusenstvi/serva-a-prislusenstvi/serva-futaba/ar01000810-servo-s3010-6-5kg-cm-0-16s-60-bb>
- [10] *Futaba S3010 Standard High Torque Servo FUTM0043* [online]. [cit. 2018-04-29]. Dostupné z: <https://www.hobbyaction.net/products/ht-standard-bb-servo>
- [11] *Power pack 3000mAh 7.2V NiMH StickPack* [online]. [cit. 2018-04-29]. Dostupné z: https://www.stavebnice4u.cz/power-pack-3000mah-7-2v-nimh-stickpack/?gclid=Cj0KCQjw2pXXBRD5ARIsAIYoEbcBsxdERANFP0FwV67EZ8kPTAU_SrStsJL6SXiyGTLDe8ITefMfbJDAAaAqB9EALw_wcB
- [12] *(TRA2925X) Traxxas NiMH baterie Car 1200mAh 7.2V iD* [online]. [cit. 2018-04-05]. Dostupné z: <https://www.mz-racing.net/tra2925x-traxxas-nimh-baterie-car-1200mah-7-2v-id/>

- [13] RAMIREZ FUENTES, Francisco, Marco TRUJILLO, Cuauhtli PADILLA a Rodrigo MENDOZA. Using Parallax TSL1401-DB Linescan Camera Module for line detection: Example code based on the MPC5604B MCU. Freescale Semiconductor Literature Distribution Center, 2011. ISBN MPC5604B.
- [14] DOČEKAL, Tomáš. *Návod pro návrh regulátoru*. VŠB - Technická univerzita Ostrava, 2016.
- [15] AS5306 Linear Sensor. Ams.com [online]. [cit. 2018-04-01]. Dostupné z: <http://ams.com/eng/Products/Magnetic-Position-Sensors/Linear-Position/AS5306>
- [16] AS5000-MR12-72 Magnetic Ring. Ams.com [online]. [cit. 2018-04-01]. Dostupné z: <http://ams.com/eng/Products/Magnetic-Position-Sensors/Magnets/AS5000-MR12-72>
- [17] *NXP Cup Ostrava Qualification Results* [online]. 2018 [cit. 2018-04-05]. Dostupné z: <https://community.nxp.com/message/997352>
- [18] BALÁTĚ, Jaroslav. Automatické řízení. Praha: BEN – technická literatura, 2004. ISBN 978-80-7300-148-3.
- [19] MAZIDI, Muhammad A., NAIMI, Sarmad, NAIMI, Sepher a Shujen CHEN. [Freescale ARM Cortex-M Embedded Programming. 2 edition. {s.l.}: MicroDigitalEd, 2016. ISBN 978-0997925982.

Seznam příloh

| | | |
|------------|--|-----------|
| Příloha A: | Základní funkce v prostředí Atolic TrueSTUDIO..... | <i>I</i> |
| Příloha B: | Funkce pro měření motorů v prostředí Arduino..... | <i>IV</i> |
| Příloha C: | CD-ROM..... | <i>IV</i> |

Příloha A: Základní funkce v prostředí Atolic TrueSTUDIO

```
void read_camera(void) {
    uint8_t i;
    set_SI();
    delay_us(10);
    set_CLK();
    delay_us(10);
    reset_SI();
    delay_us(10);
    cam_array[0] = adc_read(6);
    reset_CLK();

    for (i = 1; i < 128; i++) {
        delay_us(10);
        set_CLK();
        delay_us(10);
        cam_array[i] = adc_read(6);
        reset_CLK();
    }
}
```

Obrázek A.1: Funkce pro čtení z kamery

```
int16_t position_set(void) {
    uint8_t max = 0;
    uint8_t min = 255;
    uint8_t startL = 0;
    uint8_t startR = 127;
    int16_t position = 0;
    uint8_t i;
    uint8_t avg;

    read_camera();

    for (i = 10; i < 118; i++) {
        if (cam_array[i] > max)
            max = cam_array[i];
        else if (cam_array[i] < min)
            min = cam_array[i];
    }

    if (min < 230) {
        avg = (max + min) / 2;

        for (i = 63; i > 0; i--) {
            if ((cam_array[i] < avg) && (startL == 0))
                startL = i;
            if ((cam_array[127 - i] < avg) && (startR == 127))
                startR = 127 - i;
        }
        position = 128 - startL - startR;
    } else {
        position = 0;
    }

    return position;
}
```

Obrázek A.1: Funkce pro určení pozice

```

void servo_set(uint16_t length)
{
    if (length < MIN_PULSE)
        length = MIN_PULSE;
    else if (length > MAX_PULSE)
        length = MAX_PULSE;

    TPM1 -> CONTROLS[0].CnV = length;
}

```

Obrázek A.1: Funkce pro servomotor

```

void motors_set(uint16_t left, uint16_t right) {

    if (right < 1)
        right = 0;
    else if (right > 100)
        right = 100;

    if (left < 1)
        left = 0;
    else if (left > 100)
        left = 100;

    TPM0->CONTROLS[0].CnV = left;
    TPM0->CONTROLS[2].CnV = right;
}

```

Obrázek A.1: Funkce pro pravý a levý motor

```

void leds_set(uint8_t led, uint8_t state) {
    switch (led) {
        case 1:
            if (state == 1)
                PTB->PSOR |= (1 << 8);
            else
                PTB->PCOR |= (1 << 8);
            break;

        case 2:
            if (state == 1)
                PTB->PSOR |= (1 << 9);
            else
                PTB->PCOR |= (1 << 9);
            break;

        case 3:
            if (state == 1)
                PTB->PSOR |= (1 << 10);
            else
                PTB->PCOR |= (1 << 10);
            break;

        case 4:
            if (state == 1)
                PTB->PSOR |= (1 << 11);
            else
                PTB->PCOR |= (1 << 11);
            break;
    }
}

```

Obrázek A.1: Funkce pro LED diody

```

int main(void) {
    systick_init();
    adc_init();
    servo_init();
    camera_init();
    motors_init();
    buttons_init();
    leds_init();

    while (1) {
        if (buttons_read_A() == 1) {
            motors_set(30, 30);
            leds_set(1, 1);
        } else if (buttons_read_A() == 1) {
            motors_set(0, 0);
            leds_set(1, 0);
        } else {
            leds_set(1, 0);
            leds_set(2, 0);
        }
        if (buttons_read_B() == 1) {
            motors_set(40, 40);
            leds_set(2, 1);
        } else {
            leds_set(1, 0);
            leds_set(2, 0);
        }
        read_camera();
        delay_ms(20);
        int16_t tmp = position_set();
        tmp = 1500 + (3 * tmp);
        servo_set(tmp);
        delay_ms(20);
    }
    return 0;
}

```

Obrázek A.1: Smyčka while(1)

Příloha B: Funkce pro měření v prostředí Arduino

```
void loop() {
    while (flag == 0);
    flag = 0;
    Serial.println(cpp);
}

void encoder() {
    if (digitalRead(ENC_A) == HIGH) {
        if (digitalRead(ENC_B) == HIGH)
            count--;
        else
            count++;
    }
    else
    {
        if (digitalRead(ENC_B) == HIGH)
            count++;
        else
            count--;
    }
}

void tim1() {
    cpp = count - count_last;
    count_last = count;
    flag = 1;
}
```

Obrázek B.1: Funkce pro měření motorů

Příloha C: CD-ROM

- LAB0042_FEI_B2649_2612R041_2018.pdf - Elektronická verze bakalářské práce ve formátu PDF
- LAB0042_Atolic_kod.rar - Zdrojový kód pro řešení autonomně řízeného modelu auta v prostředí Atolic studio
- LAB0042_Arduino_měření_motoru.rar - Zdrojový kód pro měření motoru v prostředí Arduino
- LAB0042_Matlab_identifikace_motoru - Program pro identifikaci motoru
- NXP_Cup.MOV - Video záznam jízdy ze soutěže NXP Cup v národním kole